

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

WHAT IS CLAIMED IS:

1. A method of generating a stream cipher having length x bytes, the method comprising the steps of:
 - i) selecting a number n of sub-keys each having a unique non-repeating length m_n bytes;
 - ii) generating n random numbers, one for each sub-key, each having length m_n bytes;
 - iii) generating a $n+1$ st random number R ;
 - iv) set $p = \text{Mod}_{m_n}(R)$;
 - v) for each byte whose position in said n th random number is p applying a function to all n bytes to generate a value;
 - vi) concatenating said value to the end of said stream cipher;
 - vii) set $p = p+1$; and
 - viii) repeating step v), vi) and vii) until said stream cipher is x bytes in length.
2. The method of claim 1 wherein said selected length m_n of each said sub-key is a prime number.
3. The method of claim 1 wherein said selected length m_n of each said sub-key is a prime number greater than 10.

4. The method of claim 1 wherein said function applied to said n bytes of said sub-keys is the exclusive-or function.
5. The method of claim 1 comprising the further step of applying a delinearization function to said stream cipher.
6. The method of claim 5 wherein said delinearization function is a substitution cipher.
7. The method of claim 1 wherein each of said n random numbers are generated by:
 - i) generating a n + 2nd random number which is not a perfect square;
 - ii) calculating the square root of said n + 2nd random number;
 - iii) generating a n + 3rd random number;
 - iv) commencing with a digit whose position in said n + 2nd random number is calculated based on said n + 3rd random number, taking finite strings of digits sequentially and converting each said finite string into a byte;
 - v) concatenating each byte sequentially until the selected length m_n of said each of said n random numbers has been reached.
8. The method of claim 7 wherein said finite strings of digits are at least 4 digits long.

9. The method of claim 8 wherein said finite string is converted into a byte by applying a *mod* function.
10. The method of claim 7 wherein said finite string is converted into a byte by applying a *mod 256* function.
11. A computer program product for generating a stream cipher having length x bytes, said computer program product comprising a computer usable medium having computer readable program code means embodied in said medium for:
 - i) selecting a number n of sub-keys each having a unique non-repeating length m_n bytes;
 - ii) generating n random numbers, one for each sub-key, each having length m_n bytes;
 - iii) generating a $n+1$ st random number R ;
 - iv) set $p = \text{Mod}_{m_n}(R)$;
 - v) for each byte whose position in said n th random number is p applying a function to all n bytes to generate a value;
 - vi) concatenating said value to the end of said stream cipher;
 - vii) set $p = p + 1$; and
 - viii) repeating step v), vi) and vii) until said stream cipher is x bytes in length.

12. The computer program product of claim 11 wherein said selected length m_n of each said sub-key is a prime number.
13. The computer program product of claim 11 wherein said selected length m_n of each said sub-key is a prime number greater than 10.
14. The computer program product of claim 11 wherein said function applied to said n bytes of said sub-keys is the exclusive-or function.
15. The computer program product of claim 11 wherein said computer usable medium has computer readable program code means embodied in said medium for the further step of applying a delinearization function to said stream cipher.
16. The computer program product of claim 15 wherein said delinearization function is a substitution cipher.
17. The computer program product of claim 11 wherein each of said n random numbers is generated by:
 - i) generating a $n + 2$ nd random number which is not a perfect square;
 - ii) calculating the square root of said $n + 2$ nd random number;
 - iii) generating a $n + 3$ rd random number;
 - iv) commencing with a digit whose position in said $n + 2$ nd random number is calculated based on said $n + 3$ rd random number, taking

finite strings of digits sequentially and converting each said finite string into a byte;

v) concatenating each byte sequentially until the selected length m_n of said each of said n random numbers has been reached.

18. The computer program product of claim 13 wherein said finite strings of digits are at least 4 digits long.
19. The computer program product of claim 14 wherein said finite string is converted into a byte by applying a *mod* function.
20. The method of claim 5 wherein said delinearization function is a substitution cipher comprising an array of random values and in which a function is applied to two of said random values in said array to provide a substitution value.
21. The method of claim 5 wherein said delinearization function utilizes a substitution cipher comprising an array in which the values in the array are randomly repeated.
22. The method of claim 5 wherein said delinearization function utilizes a second stream cipher as a substitution cipher.

23. The method of claim 1 wherein a delinearization step is carried out during the generation of the stream cipher wherein, when applying a function to all n bytes of the sub-keys to generate a value, the position of the byte of each sub-keys to which the function is applied is selected by using the previous subkey's next byte and adding it to the offset of the current subkey.